Diffusion-based Large Language Models Survey

Chiung-Yi Tseng, Danyang Zhang, Ziqian Bi, Junhao Song†

Abstract—Diffusion-based large language models (DLLMs) have emerged as a promising alternative to traditional autoregressive architectures, notably enhancing parallel generation, controllability, and robustness across multiple modalities. Originally developed from continuous diffusion methods in computer vision, recent adaptations of DLLMs have tailored discrete diffusion processes through absorbing-state kernels, latent projections, and hybrid architectures. This survey reviews recent developments in DLLMs, beginning with their foundational concepts, including DDPM, DDIM, and their early discrete adaptations, such as mask-based, continuous-embedding, and hybrid models. We organize current methods by sampling strategy, guidance type, noise schedule, and temporal conditioning, and analyzes their efficiency, output quality, and fine-tuning. The paper also highlights key advancements: autoregressive-diffusion unification through hyperschedules, adaptive correction sampling, and efficient caching mechanisms to enhance computational performance. Besides, it explores emerging applications, such as natural language tasks, multimodal generation, and reasoning-intensive domains... These demonstrate the versatility of DLLMs. Furthermore, the paper identifies critical challenges, including adaptive sampling, scalable alignment strategies, deeper integration with pretrained language models, graph-based diffusion frameworks, and robust evaluation protocols. Finally, the paper proposes directions that could define future research in diffusion-based sequence generation.

Index Terms—Diffusion large language models, discrete denoising, large language models, latent diffusion, multimodal generation

I. Evolution of Diffusion Language Models

DIFFUSION-based language models (DLLMs) have rapidly advanced from their thermodynamic origins to versatile generators across modalities. We trace this evolution in four stages.

A. Early Foundations (2015–2020)

The original Denoising Diffusion Probabilistic Models (DDPM) introduced a Markovian forward and reverse noising chain on continuous data, establishing the core variational framework for diffusion-based generation [1]. Building on this, Denoising Diffusion Implicit Models (DDIM) generalized DDPM sampling to a non-Markovian family of deterministic or accelerated trajectories, while preserving the same training objective [2]. To handle discrete data, Hoogeboom et al. proposed Structured Denoising Diffusion Models in Discrete State-Spaces (D3PM), which uses absorbing-state kernels without explicit timestep embeddings [3].

Chiung-Yi Tseng and Danyang Zhang are with the AI Agent Lab, Vokram Group, London, United Kingdom (e-mails: ctseng@luxmuse.ai, danyang@vokram.com). Ziqian Bi is with the Department of Computer Science, Purdue University, West Lafayette, Indiana, United States (email: bi32@purdue.edu). Junhao Song is with the Department of Computing, Imperial College London, London, United Kingdom (email: junhao.song23@imperial.ac.uk).

†Corresponding author: Junhao Song (junhao.song23@imperial.ac.uk)

B. First Text-Specific Adaptations (2021–2022)

Zou et al. provided the first systematic survey of diffusion for non-autoregressive text generation, defining both discrete (mask-based) and continuous (Gaussian embedding) formulations and demonstrating their parallelism and controllability advantages over prior NAR methods [4]. Hoogeboom's D3PM work showed that absorbing-state discrete diffusion can produce coherent categorical samples without timestep inputs [3]. Concurrently, Diffusion-LM embedded Gaussian noise into token representations and outperformed earlier controllable generation models on sentiment and syntax tasks [5]. Subsequent innovations included Dieleman et al.'s continuous diffusion on one-hot vectors with specialized rounding [6], Strudel et al.'s self-conditioned embedding diffusion [7], and transformer-based ODE–diffusion hybrids such as Difformer [8] and Composable Text Controls [9]. Han et al. introduced SSD-LM, a semi-autoregressive simplex diffusion reducing denoising steps by half while preserving translation quality [10], and Yuan et al.'s SeqDiffuSeq demonstrated parallel seq2seq decoding in under 30 steps [11].

C. Hybridization Architectural Innovations (2023–2024)

The hybrid paradigm emerged with Block Diffusion, which interleaves autoregressive first-token sampling with blockwise diffusion to balance sequential fidelity and parallel speed [12]. Latent Diffusion for Language Generation projects token embeddings into a compact latent space for diffusion, then reconstructs via an AR decoder, achieving significant speedups without quality loss [13]. Energy-based approaches like DiffPO use a pretrained AR model as an energy function to reweight diffusion samples, reducing decoding error with fewer steps [14]. Speculative Diffusion Decoding treats a fast DLLM as a draft generator and an AR model as verifier, enabling near-AR quality with reduced sequential passes [15]. Task-specific optimizations, such as SSD-2's inference-time fusion of small and large diffusion LMs, further improved throughput and privacy by splitting computation [16].

D. Recent Advances and Generalization (2024–2025)

Recent work has extended DLLMs to multimodal and instruction-tuned scenarios. HybridVLA unifies vision, language, and action diffusion in a single model for embodied agent tasks [17]. Studies on instruction fine-tuning,

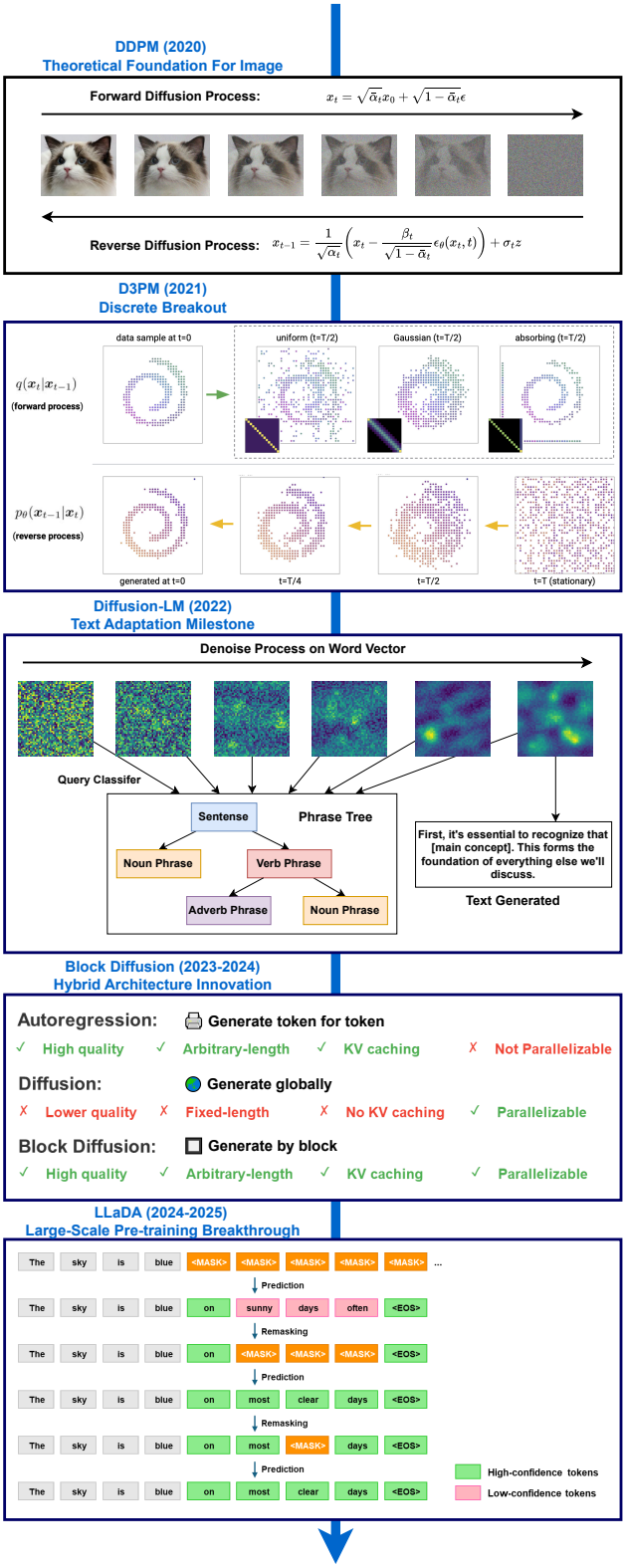


Fig. 1: Evolution timeline of major diffusion model breakthroughs (2020-2025), demonstrating the progression from foundational frameworks to advanced hybrid architectures in generative model.

such as those by Zhou et al., demonstrate that diffusion LLMs can generalize zero-shot to unseen languages and tasks after English-only SFT [18]. Comprehensive surveys highlight classification and benchmarking of DLLM capabilities across domains, marking the maturation of diffusion methods in NLP and beyond.

Together, these stages—from continuous DDPM through discrete D3PM, early NAR text diffusion, hybrid block and latent architectures, to multimodal and instruction-tuned generalists—chart the swift evolution of DLLMs into powerful, flexible generators.

II. Challenges in Diffusion-Based Text Models

DLLMs have great potential for parallelism and controllability but they also face several challenges when compared to autoregressive (AR) models. First, inference latency remains a significant hurdle: diffusion models often require dozens to hundreds of iterative denoising steps to produce coherent text. This causes much slower wall-clock inference times than the single-pass token sampling of AR decoders [12], [19].

Another limitation is the rigidity of sequence length. Many early discrete diffusion schemes assume a fixed sequence length, making variable-length or open-ended text generation difficult without additional padding, masking strategies, or complex length prediction mechanisms [12], [20].

In terms of modeling performance, diffusion-based approaches frequently underperform AR models on standard language-modeling benchmarks. This weaker likelihood performance arises from challenges in capturing sharp, multimodal discrete token distributions through gradual denoising processes [12], [14].

Furthermore, diffusion methods can struggle with global coherence, as noise is often injected uniformly or blockwise across tokens. This limits the model's ability to condition on long-range dependencies as effectively as full-attention AR decoders [12], [21].

Integrating user preferences and control signals imposes additional costs: alignment and controllability techniques such as DiffPO require extra inference-time denoising or optimization steps to steer outputs toward desired attributes, further exacerbating latency issues [22].

Mapping discrete text into a continuous diffusion framework introduces encoding and reconstruction complexity. Architectures like TEncDM rely on large projector networks to compress and reconstruct token embeddings, which can introduce errors and complicate training dynamics [23].

Basic diffusion schemes also suffer from a lack of self-correction: once a token is denoised, early mistakes cannot be revisited without rerunning the entire process. Generalized Interpolating Discrete Diffusion (GIDD) addresses this by adding fixed-point resampling iterations, but at the cost of additional computation [20].

Finally, hybrid architectures that combine AR and diffusion components—such as blockwise interpolation or hyperschedules—introduce significant engineering complexity. Coordinating two sampling paradigms, tuning noise schedules, and balancing error propagation across

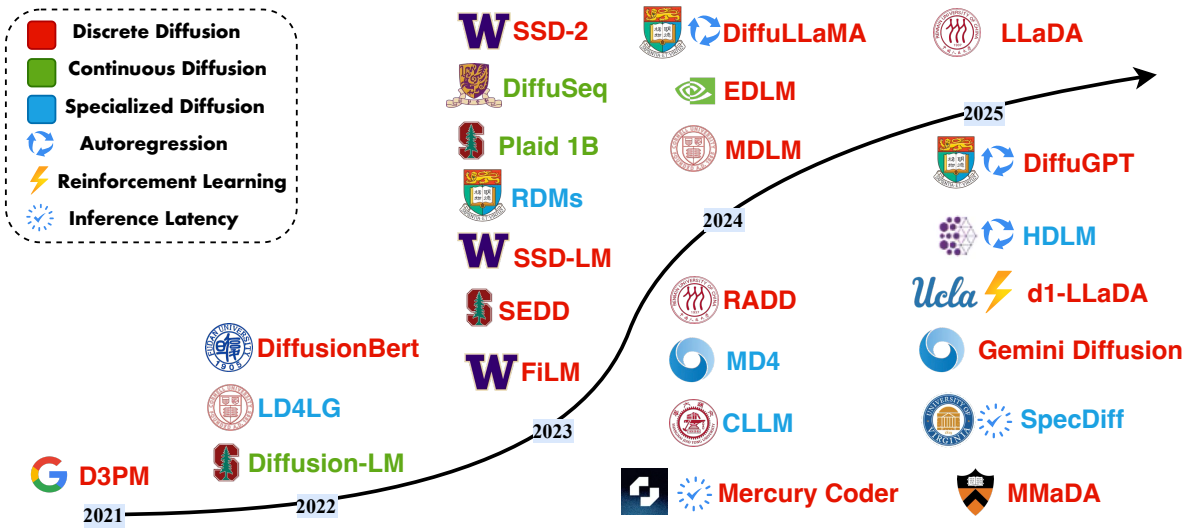


Fig. 2: Evolution of key Diffusion Language Models (DLLMs) from 2021 to 2025, annotated by architectural innovations (e.g., discrete/continuous diffusion), integration with autoregression or reinforcement learning, and inference efficiency breakthroughs.

modules can dramatically increase implementation and inference overhead [12], [21].

III. Categorization Criteria for Diffusion Language Models

We organize diffusion language models (DLLMs) along four orthogonal axes: Markovity, Guidance, Annealing Method, and Time Conditioning. The following explains the foundational works that defined each criterion.

A. Markovian vs. Non-Markovian Sampling

The distinction between Markovian and non-Markovian diffusion processes in generative modeling goes back to the original DDPM framework. The work formulates a Markovian forward and reverse noising chain [24]. The seminal work on non-Markovian diffusion processes was introduced in the Denoising Diffusion Implicit Models (DDIM) paper. The authors generalized the DDPM's Markov chain to a broader family of trajectories that retain the same training objective but allow deterministic or accelerated sampling [2].

B. Classifier-Based vs. Classifier-Free Guidance

Classifier-based guidance—reweighting diffusion samples with an external classifier's gradients—was popularized in "Improved Denoising Diffusion Probabilistic Models," showing how to trade off sample diversity and fidelity via classifier gradients during sampling [25]. Shortly thereafter, classifier-free guidance was proposed by Ho and Salimans, eliminating the need for a separate classifier by jointly training conditional and unconditional diffusion models and interpolating between their score estimates [26].

C. Annealing Method (Noise Scheduling)

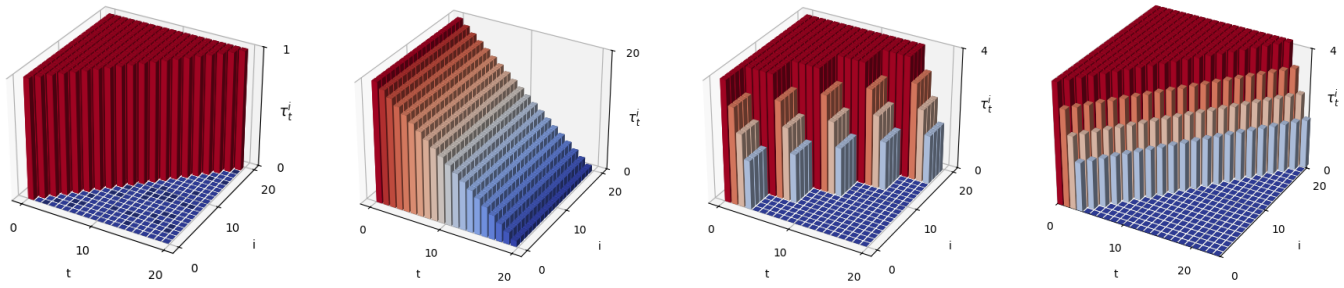
The need to choose a noise schedule for the forward diffusion process was first formalized in the original DDPM paper, which proposed a fixed variance schedule $\{\beta_t\}$ and showed how it affects both likelihood and sample quality [1]. Subsequent work ("Improved DDPM") explored alternative schedules (e.g. cosine) and adaptive variance learning for faster convergence and fewer sampling steps [25].

D. Time Conditioning vs. Time-Agnostic Kernels

In standard DDPMs, the denoiser is conditioned on an explicit timestep embedding, a mechanism first described in the DDPM framework to inform the network of the current noise level [1]. In contrast, the concept of time-agnostic diffusion—where transitions do not depend on an explicit t embedding—has been introduced in the discrete "absorbing" diffusion models, notably in Hoogeboom et al.'s D3PM work, which defines absorbing-state kernels without explicit timestep conditioning [3].

IV. Interoperability between Autoregressive and Diffusion Models

Recent theoretical and empirical advances have established fundamental connections and practical interoperability between autoregressive (AR) and diffusion-based sequence generation. Fathi et al. introduce a unified framework that treats AR and diffusion processes as points along a continuum of generation paradigms. As illustrated in Figure 3, AR models factorize sequence generation into a product of conditional token distributions, whereas diffusion models iteratively denoise from pure noise to data [21]. By modulating the noise schedule and number of denoising steps, the hyperschedules framework recovers



(a) Quenched (AR). (b) Flat annealing. (c) Block annealing. (d) Slide annealing.

Fig. 3: Fathi et al. [21] introduce $\boldsymbol{\tau}$-hyperschedules, subjecting different token positions i to different noise levels (red: high; blue: low) at different generation step t. (a) Standard AR models (e.g., GPT) determine tokens one by one, "quenching" each of them to full determination in a single step, and can be seen as an extreme case of a diffusion model. (b) Standard diffusion models (e.g., SEDD) gradually anneal all tokens independently of position. (c) Block-wise annealing for blocks of width $\omega = 4$. (d) Sliding window annealing ("smoothed" AR) with window width $\omega = 4$. These last two combine properties of both AR and diffusion.

pure AR at one extreme and pure diffusion at the other, and enables hybrid schedules that blend sequential conditioning with parallel sampling.

Ou et al. reveal an unexpected equivalence between absorbing discrete diffusion processes and arbitrary-order autoregressive models (AO-ARMs). They prove that the negative log-likelihood upper bound optimized by an absorbing discrete diffusion model corresponds exactly to the expected NLL of an AO-ARM, and they contrast reparameterized network architectures against self-conditioned denoising diffusion (SEDD/DiT) [28]. This result shows that certain discrete diffusion schemes implicitly perform off-order autoregressive factorization, suggesting new avenues for architecture design that combine the best of both approaches.

Building on these insights, Rütte et al. propose Generalized Interpolating Discrete Diffusion (GIDD), a family of masked diffusion processes that subsume prior discrete diffusion and masking-based language modeling techniques [20]. GIDD allows tokens to be noised and denoised in arbitrary patterns, enabling self-correction across positions and greater flexibility in the design of noising kernels. Theoretical analysis shows that GIDD processes interpolate between masked language models and diffusion processes, and empirical results demonstrate improved coherence, controllability, and sampling efficiency.

Together, these works demonstrate that autoregressive factorization is a special case of diffusion under particular noise schedules, and that discrete diffusion processes can emulate arbitrary token ordering schemes in AR models. This convergence lays the foundation for hybrid generation architectures that leverage the sample efficiency and strong prior modeling of AR methods alongside the parallelism and fine-grained control of diffusion processes, paving the way for more flexible and powerful sequence generation systems.

V. Mutual Knowledge Transfer between Autoregressive and Diffusion Models

Recent work has begun to explore how autoregressive (AR) and diffusion-based language models (DLLMs) can learn from one another, yielding improvements in sample efficiency, quality, and alignment.

This section examines bidirectional knowledge transfer between these paradigms, focusing on distillation techniques that enable DLLMs to leverage AR model expertise, as well as hybrid approaches that incorporate diffusion mechanisms into AR frameworks.

For DLLM training, knowledge transfer from AR models can mitigate the challenges arising from their shorter development history and reduced investment compared to their AR counterparts. DLLMs can be distilled from AR models using the same token-by-token approach employed when distilling smaller AR models from larger ones. Gong et al. present a method to in Scaling Diffusion Language Models via Adaptation from Autoregressive Models. Their approach first initializes the diffusion model's noise schedule and attention masks based on a pretrained AR checkpoint. Then the model is then fine-tuned with a combination of denoising and next-token prediction losses. The resulting DLLM inherits both the global coherence and fast convergence properties of the teacher AR model while retaining the parallel sampling advantages of diffusion. [29]. Specifically, during adaptation training, the authors employ attention mask annealing, shift operations, and a time-embedding-free architecture to reduce the gap between AR and DLLMs. 1 During sampling, $\boldsymbol{x}_T$ is initialized with all [MASK] tokens, and tokens are subsequently sampled based on the time-reversal distribution $q(\boldsymbol{x}_s|\boldsymbol{x}_t, \boldsymbol{x}_0)$ 2. The final loss at step t is computed as:

$$\mathcal{L}_t^{1:N} = \frac{1}{t}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[-\sum_{n=1}^{N}\delta_{\mathbf{x}_t^n,\boldsymbol{m}}(\mathbf{x}_0^n)^\top \log f_\theta(\mathbf{x}_t^{1:N})_n\right], \tag{1}$$

TABLE I: Comprehensive Categorization of DLLMs (Sorted by Year and Type)

Model	Markovity	Guidance	Annealing Method	Time-Embedding	Type
Diffusion-LM [5]	Non-Markovian (embedding space)	Classifier-Free	Gaussian schedule over continuous token embeddings	Yes	Continuous
Differmer [8]	Non-Markovian (ODE sampling)	Classifier-Free	Continuous-time ODE trajectories	Yes	Continuous
Composable Text Controls [9]	Non-Markovian (latent ODE)	Classifier-Free	Modular latent-space remasking with learned scheduler	Yes	Continuous
SSD-LM [10]	Semi-Markovian (simplex decoding)	Classifier-Free	Simplex annealing with half-step denoising	Yes	Discrete
D3PM [3]	Markovian (absorbing kernel)	Classifier-Free	Time-agnostic absorbing-state transitions (no explicit schedule)	No	Discrete
SeqDiffuSeq [11]	Markovian (masked denoising)	Classifier-Free	Fixed noise schedule for parallel masked denoising	Yes	Discrete
SSD-2 [27]	Semi-Markovian (fused dual-model)	Classifier-Free	Fusion annealing schedule from small to large DLLMs	Yes	Discrete
Latent Diffusion for Language Generation [13]	Non-Markovian (latent continuous)	Classifier-Free	Gaussian variance schedule in a compact latent space	Yes	Continuous
Energy-Based Diffusion LM [14]	Non-Markovian (energy sampling)	Classifier-Based (AR energy)	Standard diffusion schedule plus importance-sampling window	Yes	Continuous
HybridVLA [17]	Non-Markovian (action-space diffusion)	Classifier-Free	Continuous variance schedule over action-space diffusion	Yes	Continuous
TEncDM [23]	Non-Markovian (encoding-space)	Classifier-Free	Encoding-space diffusion with standard timestep variance	Yes	Continuous
Block Diffusion [12]	Markovian (discrete steps)	Classifier-Free	Fixed discrete noise schedule applied block-wise during denoising	Yes	Discrete
DiffPO [22]	Markovian (sentence-level denoise)	Classifier-Based (pre-trained AR energy)	Sentence-level diffusion with a predetermined noise schedule	Yes	Discrete
Speculative Diffusion Decoding [15]	Markovian (parallel draft)	Classifier-Free	Standard proposal distribution schedule for draft generation	Yes	Discrete
Your Absorbing Discrete Diffusion [28]	Markovian (absorbing kernel)	Classifier-Free	Time-agnostic absorbing Markov kernel (no explicit schedule)	No	Discrete
Generalized Interpolating Discrete Diffusion (GIDD) [20]	Markovian (masked diffusion)	Classifier-Free	Flexible mask/interpolation schedules across timesteps	Yes	Discrete

where $\delta_{x_t^n,m}$ is the indicator function that equals 1 when $x_t^n = m$ (mask token) and 0 otherwise, and $f_\theta(x_t^{1:N})_n$ represents the model output of the n-th position of the sequence. For the sampling process, the backward transition distribution conditional on $\boldsymbol{x}_0$ is defined as:

$$q(\boldsymbol{x}_s|\boldsymbol{x}_t,\boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_s)q(\boldsymbol{x}_s|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} = \begin{cases} \frac{\alpha_s-\alpha_t}{1-\alpha_t}\boldsymbol{x}_0 + \frac{1-\alpha_s}{1-\alpha_t}\boldsymbol{m} & \text{if } \boldsymbol{x}_t = \boldsymbol{m}, \\ \boldsymbol{x}_0 & \text{if } \boldsymbol{x}_t \neq \boldsymbol{m}. \end{cases} \quad (2)$$

Algorithm 1 Adaptation Training (Reproduced from [29])

Input: network f_θ initialized by existing models, training corpus $p_{data}(\boldsymbol{x}_0^{1:N})$, mask token $\boldsymbol{m}$
Output: model parameters θ
repeat
  Draw $\boldsymbol{x}_0^{1:N} \sim p_{data}$ and set labels $\leftarrow \boldsymbol{x}_0^{1:N}$
  Sample $t \sim \text{Uniform}(0,1)$
  Sample $\boldsymbol{x}_t^{1:N} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$
  Anneal the attention mask attn_mask
  Forward pass: logits $\leftarrow f_\theta(\boldsymbol{x}_t^{1:N})$ with attn_mask
  Right shift logits by one position ▷ see Eq. 1
  $\mathcal{L}_t = \frac{1}{t}\delta_{x_t,m} \cdot \text{CE}(\text{logits}, \text{labels})$
  Backpropagate using $\mathcal{L}_t$ and update θ
until convergence

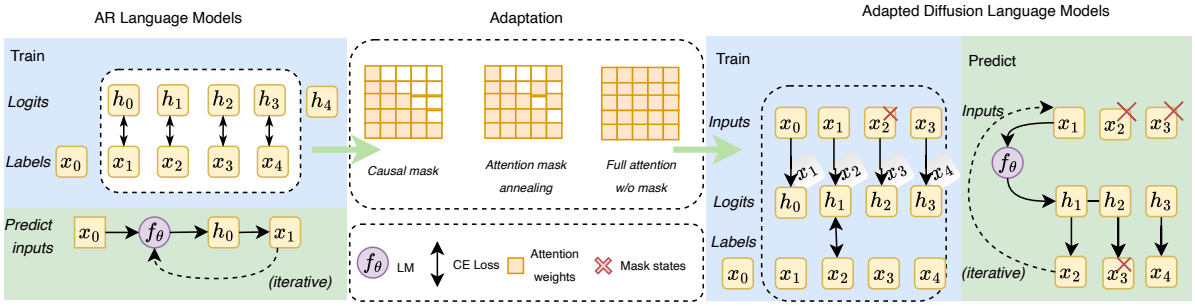


Fig. 4: The overview of Gong et al.'s [29] approach to adapt autoregressive (AR) models to diffusion models. Left: The shift operation in AR models enables the output layer h_i to approximate the distribution of next tokens x_{i+1} in hidden representations through the cross entropy (CE) loss. Middle: Gradually removing the causal mask during training eventually makes the model bidirectional. Right: Inside the diffusion models, shifting the logits to compute the loss with the next token (i.e., the loss on h_i would be concerning x_{i+1}), while perceptually, the diffusion models are still functioning as recovering the original signals (since h_i corresponds to x_{i+1} in AR loss).

Algorithm 2 Sampling (Reproduced from [29])

Input: Trained diffusion model f_θ, sampling algorithm τ, mask token $\boldsymbol{m}$, start token $\boldsymbol{s}$
Output: generated sample $\boldsymbol{x}_0$
Initialize $\boldsymbol{x}_T^{1:N} \leftarrow \boldsymbol{m}$
for $t = T, \ldots, 1$ do
  logits $\leftarrow f_\theta(\boldsymbol{x}_t^{1:N})$
  $\tilde{\boldsymbol{x}}_0^{1:N} \sim \text{Categorical}(\tau(\text{logits}))$
  for $n = 1, \ldots, N$ do
    $\boldsymbol{x}_{t-1}^n \leftarrow q(\boldsymbol{x}_{t-1}^n \mid \boldsymbol{x}_t^n, \tilde{\boldsymbol{x}}_0^n)$ ▷ Eq. 2
  end for
  Right shift $\boldsymbol{x}_{t-1}^{1:N} \leftarrow [\boldsymbol{s}, \boldsymbol{x}_{t-1}^{1:N-1}]$
end for
return $\boldsymbol{x}_0^{2:N}$

Energy based diffusion offers an alternative distillation approach: Xu et al. demonstrate that by employing a pretrained AR model as the energy function within a diffusion framework, one effectively distills the AR model's token probabilities into a diffusion proposal distribution. At inference time, samples drawn in parallel from the diffusion model are reweighted or rescored by the AR energy function, correcting decoding errors and achieving AR level quality with substantially fewer sequential steps [14].

These distillation techniques also enhance speculative decoding: when the diffusion draft model has been distilled from an AR teacher, its proposals during speculative sampling exhibit significantly higher accuracy, resulting in reduced rejection rates and improved overall throughput compared to undistilled drafts [15].

Conversely, AR models can incorporate diffusion style representations. Lovelace et al. repurpose the encoder-decoder latent space of a pretrained AR model to learn a high-dimensional diffusion process that captures token-to-token correlations beyond the standard AR factorization. By interleaving diffusion-based latent refinement steps within the AR decoding loop, the hybrid system achieves superior diversity and controllability while maintaining AR fluency [13].

Together, these research directions establish a bidirectional bridge: AR models can seed and guide efficient DLLM training through distillation, while DLLM mechanisms can enrich AR decoders with parallel refinement capabilities. This mutual transfer of inductive biases promises novel hybrid architectures that effectively combine the strengths of both paradigms.

VI. Collaboration between Diffusion-based and Autoregressive Models at Inference Time

Recent work has shown that diffusion-based language models (DLLMs) and autoregressive (AR) models can be combined in complementary ways to leverage the strengths of both paradigms. For example, HybridVLA interleaves an AR component for text and image generation with a diffusion component operating on the action space of an embodied agent, allowing a single model to generate rich multimodal descriptions and execute coherent action plans [17]. In Block Diffusion, fixed-size blocks of tokens are generated by first sampling the initial token autoregressively, then denoising the remainder of the block in parallel via a DLLM, achieving a balance between sequential decoding speed and parallel sampling efficiency [12].

DiffPO applies a diffusion-style denoising step at inference time to align AR-generated sentences with learned human "preference" vectors, reshaping outputs post hoc without retraining the underlying AR model and improving both fluency and alignment metrics [22]. Similarly, Latent Diffusion for Language Generation projects token embeddings into a compact latent space via a compression network and reconstructs tokens through an AR decoder, speeding up sampling while preserving generation quality [13].

In an energy-based approach, Energy-Based Diffusion Language Models train a diffusion process to match a target AR distribution, enabling "plug and play" sampling from pretrained AR models via energy gradients and bridging sample efficiency with expressive power [14]. Building on this idea, Speculative Diffusion Decoding uses a fast DLLM as a draft generator to propose multiple continuations in parallel, which are then scored or filtered

by an AR model to achieve near AR quality with fewer sequential steps [15].

DDPT: Diffusion-Driven Prompt Tuning uses a DLLM to generate or refine prompts in latent space for a large language model focused on code generation; the prompt is then passed to an AR code generator, and the overall loss computed against ground truth code tunes the prompt space via diffusion [30]. On the theoretical side, a Unified Hyperschedules Framework demonstrates that autoregression arises as a special case of diffusion under extreme noise schedules; by modulating noise levels and step counts, one can smoothly interpolate between pure AR and diffusion, or combine them to harness both methods' advantages [21].

TEncDM operates diffusion not on raw tokens but in the continuous encoding space of a frozen LLM encoder–decoder, using cross attention for self-conditioning. This leverages powerful transformer representations while accelerating refinement through parallel diffusion sampling [23].

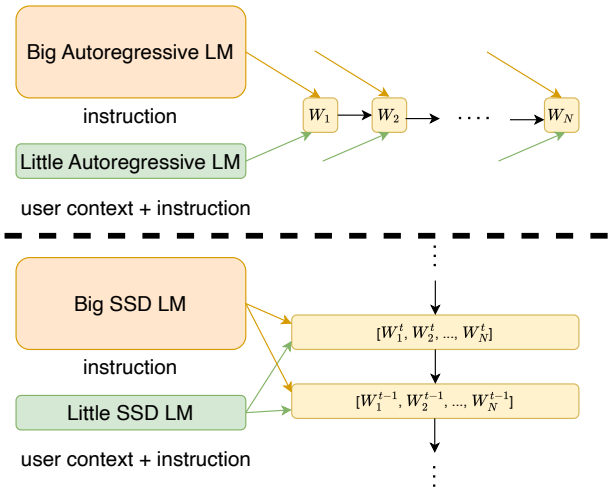


Fig. 5: David helps Goliath [16]: Inference-time collaboration between large and small models. AR models decode token-by-token, while diffusion models refine token blocks iteratively with bidirectional contexts.

Finally, David helps Goliath demonstrates DLLM's mutual collaboration advantage over autoregressive counterparts: DLLMs' iterative generation process enables them to reference bidirectional context, therefore, it is easier for different DLLMs to collaborate at the sequence level and yield better quality. Inspired by AR counterparts, the authors incorporate the logits-averaging method to generate a block of tokens at each diffusion step for expert model θ_{core} and user model θ_{user}. Similar to AR counterparts, to increase the pointwise information exchange between the expert generated data and the conditioned generation based on the instruction, a contrastive term θ_{user} without input D_{user} is added. [16]

$$
\begin{aligned}
\boldsymbol{w}^{c:c+B}_{\text{core-logits},t} &= \text{logits}_{\theta_{\text{core}}}(\boldsymbol{w}^{c:c+B} \mid \boldsymbol{w}_{\text{inst}}, \boldsymbol{w}^{<c}, \tilde{\boldsymbol{w}}_t^{c:c+B}) \\
\boldsymbol{w}^{c:c+B}_{\text{user-logits},t} &= \text{logits}_{\theta_{\text{user}}}(\boldsymbol{w}^{c:c+B} \mid D_{\text{user}}, \boldsymbol{w}_{\text{inst}}, \boldsymbol{w}^{<c}, \tilde{\boldsymbol{w}}_t^{c:c+B}) \\
\boldsymbol{w}^{c:c+B}_{\neg\text{user-logits},t} &= \text{logits}_{\theta_{\text{user}}}(\boldsymbol{w}^{c:c+B} \mid \boldsymbol{w}_{\text{inst}}, \boldsymbol{w}^{<c}, \tilde{\boldsymbol{w}}_t^{c:c+B}) \\
\boldsymbol{w}^{c:c+B}_{\text{logits},t} &= (1-\lambda_{\text{user}})\boldsymbol{w}^{c:c+B}_{\text{core-logits},t} \\
&\quad + \lambda_{\text{user}}(1+\alpha)\boldsymbol{w}^{c:c+B}_{\text{user-logits},t} - \lambda_{\text{user}}\alpha\boldsymbol{w}^{c:c+B}_{\neg\text{user-logits},t}
\end{aligned}
$$

Together, these approaches illustrate a rich design space for hybrid generation: blockwise and latent space diffusion, post-hoc preference alignment, speculative drafting, prompt tuning, and unified theoretical frameworks. All pointing toward future models that fluidly integrate diffusion and autoregression within a single system.

VII. Inference Speed of Diffusion Language Models

In David helps Goliath, the authors find the generation no longer updates for the last 40% of the inference time. They incorporate an early-stop strategy: halting the small model from generation based on empirically determined time range: t = 0.4T, the pipeline reduces the total number of denoising steps by 40% without sacrificing output quality. [16]

Building on ideas from autoregressive speculative decoding, Speculative Diffusion Decoding uses a fast DLLM as a draft generator and an AR model as the verifier. The diffusion draft draws k tokens in parallel and the AR target then confirms or corrects them, yielding substantial speedups even without fine-tuning the diffusion draft [15]. This two-stage process highlights how parallel sampling can be combined with selective sequential verification to accelerate decoding.

Self-conditioned discrete diffusion processes can match autoregressive latency through parallel sampling. In their work on SEDD, Lou et al. demonstrate that around 100 denoising steps suffice to match AR inference time, and by removing KV-cache dependencies, throughput can increase by 4–6× [31]. This result highlights the potential of batch parallelism and cache-free sampling in reducing the speed gap between diffusion and autoregressive models.

Task-specific optimizations also yield notable gains. The Discrete Diffusion Language Model for Efficient Text Summarization employs tailored noise schedules and reduced-step denoising to outperform both AR and continuous-diffusion baselines on wall-clock decoding time [32]. By focusing computation on the most informative tokens, the efficiency of summarization is greatly enhanced.

Reparameterization of discrete diffusion versus continuous dynamics has a significant impact on convergence speed. Zheng et al. analyze how continuous diffusion over token embeddings converges slowly—meaningful tokens only emerge after hundreds or thousands of iterations—while reparameterized discrete diffusion achieves coherent text generation in tens of steps, explaining much of the observed speed disparity [19].

In Energy-Based Diffusion Language Models, a pretrained AR model serves as an energy function within the diffusion sampler. By applying importance sampling on late timesteps, required denoising steps can be reduced while maintaining AR-level generation quality under the same latency budget [14]. This approach illustrates how energy-based rejection sampling can accelerate diffusion inference.

Finally, Your Absorbing Discrete Diffusion removes time-conditioning in its Markov kernel to enable KV-style caching of intermediate predictions, further speeding up sampling compared to standard absorbing diffusion formulations [28]. This caching mechanism highlights how architectural modifications can alleviate the overhead of iterative denoising.

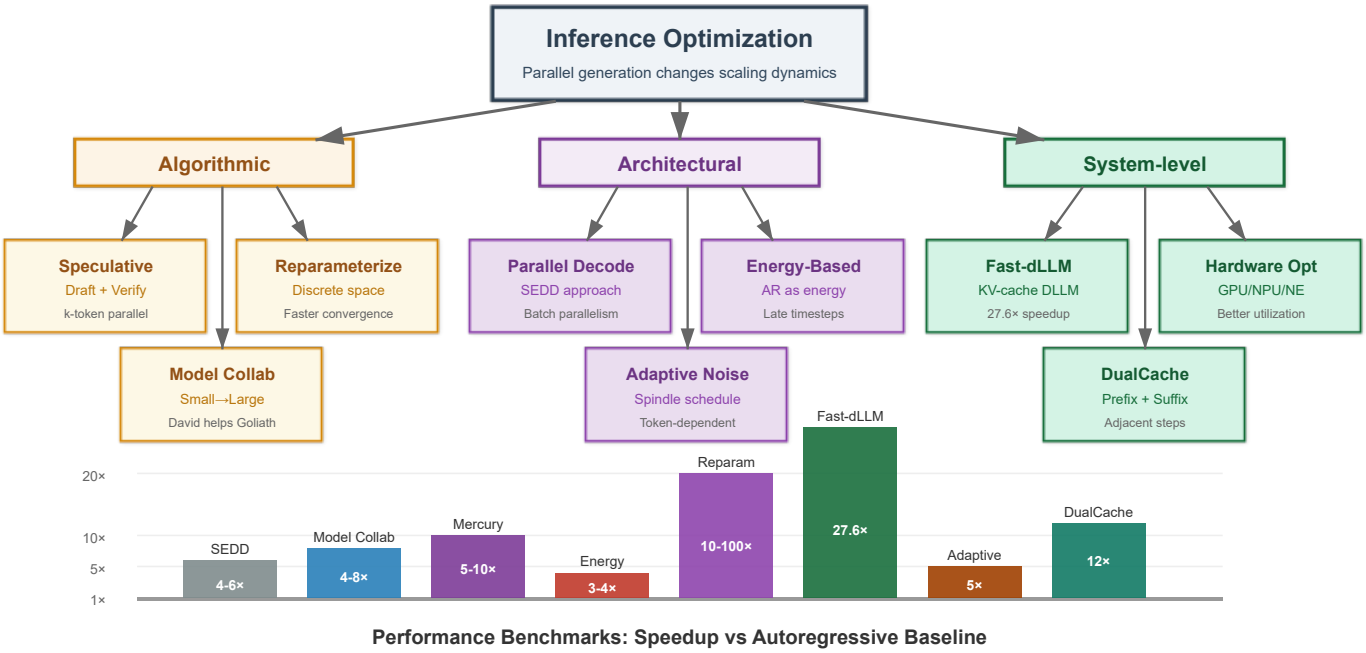


Fig. 6: DLLM Inference Speed Optimization Techniques

VIII. Impact of Sampling Techniques on Performance

Sampling strategies are central to the trade-off between quality, speed, and stability in diffusion-based text generation. In energy-based diffusion models, a pretrained autoregressive (AR) model serves as an energy function to reweight samples drawn from the diffusion proposal distribution. The method restricts the importance sampling to a window of late diffusion timesteps and significantly reduces parallel decoding error. It enables high-fidelity outputs with significantly fewer denoising iterations, thereby reducing overall wall-clock time. Complete MCMC sampling remains infeasible in the high-dimensional token space. Therefore, importance sampling remains the preferred alternative [14].

Furthermore, masked diffusion models can accelerate inference by skipping redundant noise levels. In "Simple and Effective Masked Diffusion Language Models," the authors introduce an Efficient Ancestral Sampling scheme that dynamically omits specific intermediate timestamps during denoising, reducing the number of functional calls to the denoiser without degrading output quality [33].

Recently, the Large Language Diffusion Models (LLaDA) framework employs two complementary remasking strategies to concentrate computation on uncertain tokens. First, low-confidence remasking re-noises only those tokens whose predicted confidence falls below a threshold, focusing denoising steps where they matter most. Second, a semi-autoregressive remasking divides the sequence into blocks that are generated left-to-right but sampled in parallel within each block, achieving a balance between sequential coherence and parallel throughput [18].

Generalized Interpolating Discrete Diffusion (GIDD) adds a self-correction iteration after full denoising: the model resamples tokens based on its likelihood estimates, committing the highest-scoring changes until convergence. The fixed-point refinement step mitigates error propagation from early timesteps, thus yields more coherent sequences without requiring additional training [20].

To sum up, these sampling innovations: importance sampling windows, efficient ancestral skipping, confidence-based remasking, and self-correction demonstrate the careful scheduler and sampler design can significantly improve both the speed and quality of diffusion-based text generation.

IX. Fine-Tuning of Diffusion Language Models

Fine-tuning diffusion language models (DLLMs) extends their flexibility and task adaptability, drawing inspiration from autoregressive (AR) LLM fine-tuning techniques. Recent work demonstrates several viable pathways.

In David helps goliath, the authors report one of the first instances of DLLM fine-tuning by leveraging the DOLLY dataset (15K human-collected instructions and responses) to adapt their SSD-2 diffusion model for downstream tasks. The authors show the finetuned model is better at collaboration than autoregressive counterpart thanks to the bidirectional context reference during ensemble [16].

The Mercury Diffusion Language Model technical report explores alignment-focused fine-tuning strategies: reinforcement learning from human feedback (RLHF) and direct preference optimization (DPO) have been used to guide the diffusion denoiser towards preferred outputs. By integrating these feedback signals at inference-time trajectories, the model rapidly aligns to user-specified functions, suggesting that DLLMs can match or exceed AR alignment speed [34].

Supervised fine-tuning (SFT) also proves effective. In Large Language Diffusion Models (LLaDA), the authors demonstrate conventional instruction-driven SFT on a diffusion backbone, showing scalable improvements in generative accuracy across a range of tasks. By framing instructions as conditioning masks in the denoising process, DLLMs achieve comparable zero-shot performance to AR LLMs post-finetuning [18].

Beyond standard SFT, Instruction Fine-Tuning further enhances generalization. Diffusion Language Models Can Perform Many Tasks with Scaling and Instruction-Finetuning shows instruction-tuned diffusion models generalize to unseen languages and tasks without additional data. E.g., German text generation emerges zero-shot after English only fine-tuning and highlighting the strong transfer capacity of DLLMs [35].

These studies reveal a promising interpolation between AR and diffusion fine-tuning. Parameter-efficient adapters and instruction masks can be seamlessly adapted to the denoising

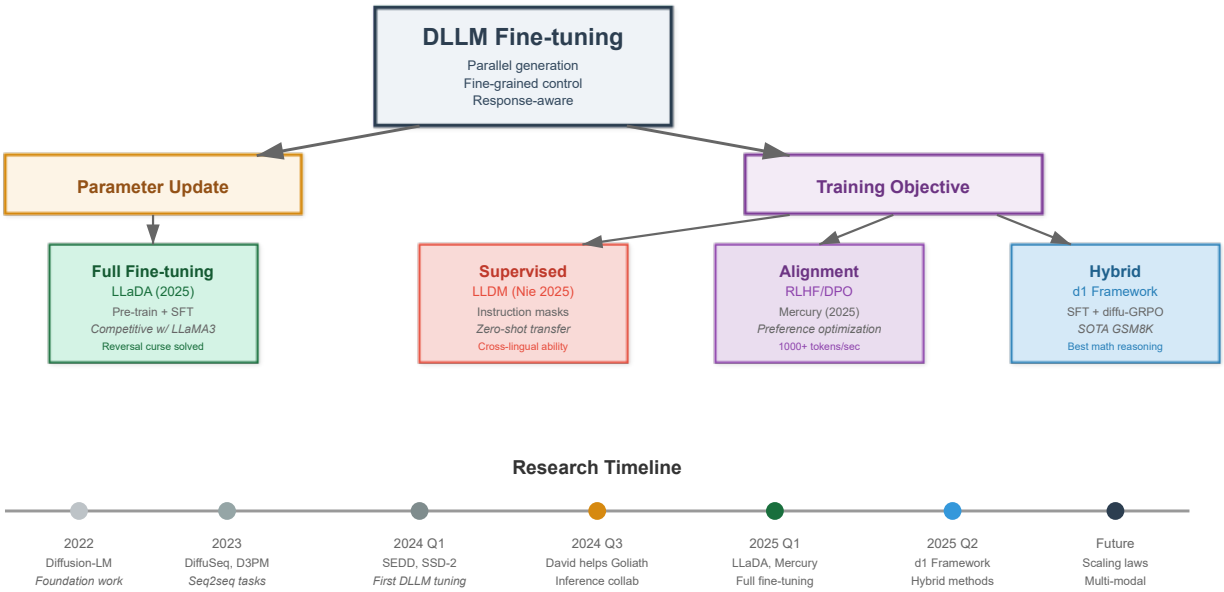


Fig. 7: DLLM Fine-tuning Methods: Taxonomy and Evolution

architecture; this enables rapid specialization and alignment. Potential future work includes integrating LoRA - low-rank adaptation techniques to reduce tuning overhead and explore hybrid schedules that jointly adjust AR and diffusion components during fine-tuning.

X. Multimodality and Reasoning Capabilities of Diffusion Language Models

Recent work has begun to expand diffusion language models (DLLMs) beyond pure text generation into the realms of multimodal understanding and complex reasoning. In the multimodal domain, Diffusion Language Models Can Perform Many Tasks with Scaling and Instruction-Finetuning demonstrates that a single discrete-diffusion backbone can be extended to process image inputs and answer visual questions. By following the two-phase training paradigm of LLaVA—first interleaving visual and textual encodings in a frozen vision encoder, then instruction-fine-tuning the diffusion decoder—DLLMs attain zero-shot performance on vision-language benchmarks without specialized cross-modal architectures [35]. This result highlights DLLMs' ability to treat images as another "language" of tokens, leveraging the random-masking denoising process to fuse visual and linguistic representations seamlessly.

On the reasoning front, d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning introduces a two-stage post-training framework—supervised fine-tuning on high-quality reasoning traces followed by a novel policy gradient method (diffu-GRPO)—to imbue masked DLLMs with stepwise planning capabilities. Remarkably, as sequence lengths exceed 512 tokens, the model begins to exhibit self-correction and backtracking behaviors, mirroring the "chain-of-thought" processes seen in autoregressive counterparts, yet without relying on a fixed generation order. Furthermore, instruction-fine-tuned DLLMs are shown to conform their generative steps to a topological sort of the underlying causal graph—first generating premises, then formulas, then conclusions—underscoring their inherent advantage in modeling non-linear dependencies over unidirectional AR decoders [36].

The key innovation of d1 lies in adapting Group Relative Policy Optimization (GRPO) for the DLLM's architecture. Traditional policy gradient methods encounter challenges here due to DLLMs non-autoregressive nature. To address this, the authors develop diffu-GRPO, which modifies the standard GRPO objective to handle partially masked sequences. To compute the log-probability of each token o in the sequence in the diffu-GRPO framework, for a given prompt q, a perturb process is executed on it where each token randomly gets masked with probability p_{mask}, generating a masked prompt q'. Then a one step unmasking process is executed to obtain the estimation of per-token log-probability $\log \pi_\theta(o^k|q)$, $1 \le k \le |o|$. The advantage for token k in response o_i is computed using the group-relative approach:

$$A_i^k(\pi) = r_i(\pi) - \text{mean}\left(\{r_j(\pi)\}_{j=1}^G\right), \quad 1 \le k \le |o_i|, \tag{3}$$

where $r_i(\pi)$ represents the reward for response i, and G is the total number of responses sampled from the current policy. The optimization objective combines this advantage calculation with policy gradient updates specifically designed for masked sequences. The diffu-GRPO loss function incorporates both the policy improvement term and KL divergence regularization:

$$\begin{aligned}\mathcal{L}_{\text{diffu-GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D},\, q'\sim\text{masking}(q),\\ o_1,\ldots,o_G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|} \\ \min\left(\frac{\phi^{\pi_\theta}(o_i^k \mid q')}{\phi^{\pi_{\theta_{\text{old}}}}(o_i^k \mid q')}A_i^k,\right. \\ \left.\text{clip}\left(\frac{\phi^{\pi_\theta}(o_i^k \mid q')}{\phi^{\pi_{\theta_{\text{old}}}}(o_i^k \mid q')}, 1-\varepsilon, 1+\varepsilon\right)A_i^k\right) \\ -\beta\, D_{\text{KL}}\Big[\phi^{\pi_\theta}(\cdot \mid q') \,\Big\|\, \phi^{\pi_{\text{ref}}}(\cdot \mid q')\Big]\Bigg],\end{aligned} \tag{4}$$

where $\phi^{\pi_\theta}(o^k \mid q')$ and $\phi^{\pi_\theta}(o \mid q')$ denote the estimated per-token and sequence probabilities for π_θ, and β controls the strength of the KL divergence penalty. The algorithm of diffu-GRPO is summarized in Algorithm 3.

By abstracting both images and complex logical steps as masked tokens in a unified diffusion process, these frontier studies reveal that DLLMs can serve as generalist engines for multimodal perception and multi-step reasoning. Future work will undoubtedly refine these training recipes and explore more modality integrations and "knowledge transfer" of different

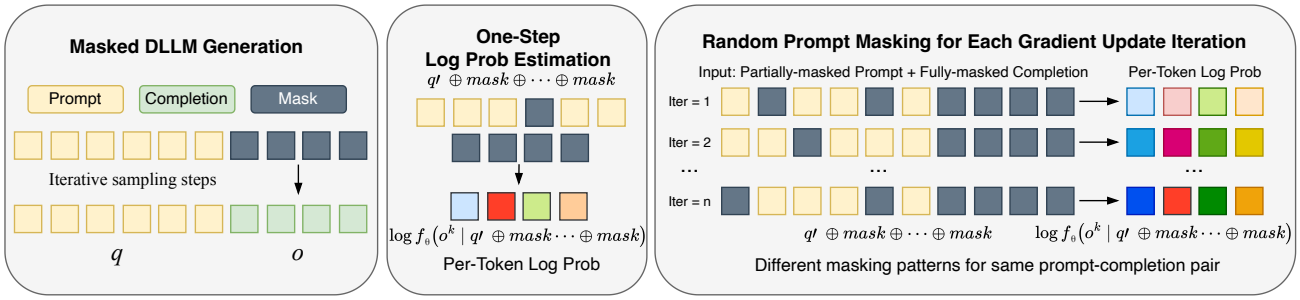


Fig. 8: Log Probability Estimation in diffu-GRPO. First the completion o is generated from prompt q using full diffusion sampling (left). Then token-level log probabilities are computed using a single forward pass for each masking pattern (mid). The log-probability from one-step unmasking is used as the estimation method. During policy gradient updates, a random masking pattern is applied to the prompt to create q', while the completion stays fully masked (right). The color gradients in per-token log probabilities show that different masking patterns give different estimates of token-level log probabilities. This works as a regularization method for policy optimization, allowing more gradient updates per batch. This approach reduces the number of online generations needed for RL training.

Algorithm 3 diffu-GRPO: Policy Gradient Optimization for Masked dLLMs (Reproduced from d1 [36])

Require: Reference model π_{ref}, prompt distribution $\mathcal{D}$, number of completions per prompt G, number of inner updates μ, prompt token masking probability p_{mask}
Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$
while not converged do
  $\pi_{\text{old}} \leftarrow \pi_\theta$
  Sample a prompt $q \sim \mathcal{D}$
  Sample G completions $o_i \sim \pi_{\theta_{\text{old}}}(\cdot|q)$, $i \in [G]$
  For each o_i, compute reward r_i and advantage $A_i^k(\pi_{\theta_{\text{old}}})$ using Eq. 3
  for gradient update iterations $n = 1, \ldots, \mu$ do
    $q' \leftarrow$ randomly mask tokens of prompt p with probability p_{mask}
    For $\pi_\theta, \pi_{\theta_{\text{old}}}, \pi_{\text{ref}}$, estimate log-probabilities of o_i given q'
    Compute diffu-GRPO objective (4) and update π_θ by gradient descent
  end for
end while
return π_θ

modalities between AR models and DLLMs.

XI. Evaluation of Diffusion Language Models

Evaluating diffusion language models (DLLMs) requires metrics that capture both generative quality and the distinctive inference characteristics of diffusion processes. We discuss common benchmarks and their limitations when applied to DLLMs.

Language modeling. To compare token prediction accuracy with autoregressive (AR) baselines, DLLMs often report perplexity on datasets like WikiText-103 and Penn Treebank. For example, Block Diffusion and Reparameterized discrete models approach AR perplexity after sufficient training steps [12], [19]. However, perplexity assumes a left-to-right factorization and ignores DLLM's parallel sampling and iterative refinement. This potentially underestimates diffusion models optimized for inference speed. Complementary metrics such as BERTScore have been proposed to evaluate contextual similarity beyond token-level likelihood [37], yet these do not reflect generation latency.

Sequence-to-sequence tasks. Machine translation and summarization evaluations use BLEU and ROUGE scores on WMT and CNN/DailyMail benchmarks [38], [39]. Discrete Diffusion for Summarization achieves ROUGE comparable to AR systems with fewer denoising steps [32]. Besides, COMET provides a learned evaluation that correlates better with human judgments on translation [40]. Diversity metrics such as distinct-n and Self-BLEU quantify the variety of generated outputs, which is important for diffusion modelsparallel proposals [41].

Open-ended generation. Distributional metrics such as MAUVE measure the divergence between model and human text distributions [42]. Meanwhile, diversity measures (distinct-1/2) assess intra-sample variety. MAUVE better reflects human preference than perplexity, but it overlooks diffusion's multi-step denoising trajectories and interactive editing capabilities. Human evaluations, preferences or side-by-side comparisons, are still essential yet are infrequent due to cost.

Reasoning and instruction following. Benchmarks, such as GSM8K, CommonsenseQA, and StrategyQA assess multi-step and commonsense reasoning. Instruction-fine-tuned DLLMs demonstrate zero-shot performance on unseen tasks, matching AR LLMs in reasoning accuracy [36]. BigBench Hard tasks reveal areas where diffusion planning may excel [43]. However, some reasoning benchmarks focus on final answers and ignore DLLM's intermediate planning and self-correction behaviors.

Multimodal and embodied tasks. Models such as HybridVLA are evaluated on VQA accuracy, COCO caption CIDEr, METEOR, and SPICE scores [17], [44]. Although, these benchmarks test cross-modal alignment, they do not capture DLLM's capacity for iterative visual refinement or privacy-preserving on-device drafts.

Robustness and calibration. Adversarial and stress tests—e.g. perturbed prompts or style shifts, measure model stability under input variation [45]. Calibration metrics such as expected calibration error (ECE) evaluate confidence alignment, relevant for diffusion's energy-based sampling, however it is rarely reported [46].

Inference speed and efficiency. Real-world evaluation should include wall-clock latency and throughput. Self-Conditioned Discrete Diffusion (SEDD) matches AR latency at 100 steps and attains 4–6× higher batch throughput by removing KV-cache dependencies [31]. Energy-based importance sampling reduces denoising iterations, achieving AR-level quality under the same time budget [14]. Most DLLM works, however, still report only step counts rather than end-to-end latency metrics.

While perplexity, BLEU/ROUGE, MAUVE, and other distributional or learned metrics offer useful baselines, they often underrepresent diffusion's parallelism, controllability, and iterative refinement. We advocate for latency-aware benchmarks,

intermediate-step quality tracking, and interactive tasks that leverage self-correction and multi-token proposals to fully characterize DLLM capabilities.

XII. Applications of Diffusion Language Models

Diffusion language models (DLLMs) have grown beyond simple text generation. When we regard specialized data—such as protein sequences, molecular graphs, or genomic DNA—as distinct "languages," DLLMs demonstrate remarkable versatility across both scientific and multimodal fields.

In text-to-video generation, researchers have successfully combined large language models with diffusion priors to generate videos base on the natural-language instructions. For example, in "The Best of Both Worlds: Integrating Language Models and Diffusion Models for Video Generation," the authors show how prompt-conditioned diffusion transformers can generate successive frames that remain temporally consistent and closely aligned with the user's text prompt [47].

DLLMs have also been applied to protein and molecule design. In DiffSDS, protein backbone inpainting is formulated as a masked diffusion process over torsion angles. This enables the model to complete structures accurately under defined geometric constraints [48]. DPLM-2 employs two separate tokenizers—for amino-acid sequences and structural motifs—and jointly denoises both representations to propose novel protein scaffolds [49]. In the domain of small-molecule discovery, Constrained Discrete Diffusion (CDD) enforces chemical valence and substructure rules at every denoising iteration. The technique ensures that the generated compounds are both chemically valid and sufficiently novel for drug and materials research [50].

Beyond proteins, DLLMs have been extended to genomic sequence modeling. Simple and Effective Masked Diffusion Language Models (MDLM) is a new way to generate text by gradually filling in masked words. It trains an encoder-only model with a mix of classic masked language losses, then samples text in a few semi-autoregressive steps. On benchmarks like LM1B and OpenWebText, MDLM cuts diffusion model perplexity close to standard autoregressive methods. [33].

In text summarization, CrossMamba perform blockwise denoising over entire documents, achieving faster decoding speeds and better content preservation than typical autoregressive summarization on Gigaword andCNN/DailyMail datasets [32]. Moreover, DLLM-based classifiers can maintain an author's original stance—particularly in political news—by applying preference-guided denoising at inference time [51].

The applications extend into robotics. The HybridVLA model fuses diffusion-generated action plans with autoregressive textual descriptions, enabling an embodied agent to both plan and verbally explain its actions in a coherent manner [17].

For privacy-sensitive inference, the David helps Goliath framework partitions computation: a lightweight diffusion model operates locally on the user's device and a remote expert model. The generation on the local diffusion model only defers to a remote expert model when necessary. In this way, user data remains confidential while the overall generative performance is preserved [16].

DLLMs have further been adapted for seq2seq tasks, such as machine translation and data-to-text conversion, through DiffuSeq's parallel masked denoising pipeline [52]. In alignment-based editing,methods like DiffPO apply targeted denoising to achieve controlled sentiment or stylistic transformation in text [22]. There are even emerging models for sign language production, where sequences of gesture tokens are generated from textual input using discrete diffusion processes [53].

By abstracting structured data as discrete languages, DLLMs unlock new possibilities in scientific discovery, multimedia creation, and human–machine interaction—showing that the same underlying principles can be tailored to a wide array of domains.

XIII. Future Research Directions

Despite that diffusion language models (DLLMs) have made great progress in text, multimodal, and scientific domains, several challenges and promising directions are yet to be explored.

Adaptive and budgeted sampling seem promising. Although Speculative Diffusion Decoding [15] and importance sampling windows in energy-based samplers [14] yield speedups, DLLMs still rely on tens to hundreds of denoising steps. Future work may explore learned noise schedules that dynamically allocate iterations based on token uncertainty; implement budgeted diffusion, in which a global compute or latency budget enables early stopping. Continuous-time samplers such as DDIM [2] that can execute sampling process with adaptive step-size control, may offer further acceleration while preserving sample quality.

Alignment and controllability remain another key issue. Initial alignment techniques: RLHF and DPO in the Mercury report [34] and DiffPO [22], demonstrate feasibility on small models, but scaling to large diffusion architectures demands parameter-efficient adapters (e.g., LoRA) tailored for denoising networks. Moreover, integrating classifier-free guidance [26] into diffusion samplers could enable fine-grained control without having external energy networks.

Hybrid diffusion–autoregressive paradigms can combine the strengths of both worlds. Block Diffusion [12] and TEncDM [23] illustrate how AR and diffusion models cooperate. Future research may investigate joint training objectives, that blend maximum-likelihood AR losses with score-matching. Build on this, researchers may develop mixture-of-sampling experts that choose between AR and diffusion per token or block based on context and compute constraints.

Multimodal and structured generation are ripe for expansion. Building on HybridVLA [17], we see modality-agnostic architectures having the capability of unified text, vision, audio, and graph diffusion. In scientific domains, graph-structured diffusion processes for molecules [20] and proteins [48] can benefit from self-correction techniques exemplified by GIDD [20] to enforce structural validity.

The interactive nature of diffusion sampling suggests novel human-in-the-loop applications. Live editing interfaces can allow users to re-mask and refine tokens during generation,while cascaded refinement pipelines can progressively enhance outputs via coarse-to-fine diffusion stages. Such interactive paradigms can leverage DLLM's parallel proposals for rapid user feedback.

Understanding and interpreting diffusion models across languages remains under explored. We encourage research about information-theoretic analyses: how noise schedules impact modeling capacity and diversity. For theoretical work, researchers should focus on bridging diffusion and autoregression via the hyperschedules framework [21]. These works will bring clarity on when diffusion models approximate AR counterparts and guide the design of efficient hybrid samplers.

Finally, evaluation benchmarks must consider DLLM's unique attributes. Beyond quality metrics (perplexity, BLEU, ROUGE, MAUVE), we propose three additional metrics: Latency-aware quality curves tracks performance across denoising steps. Interactive editing benchmarks measures mid-generation corrections. Privacy and robustness tests evaluates on-device draft sampling versus server-side verification. These new assessments will align research objectives with diffusion models' core advantages.

XIV. Conclusion

Our survey has outlined the evolution of DLLMs from foundational Markovian designs such as DDPM and D3PM to contemporary non-Markovian, hybrid, and energy-based models. These models establish new connections between diffusion and autoregression. We categorized these approaches across four axes: Markovity, guidance, annealing method, and time-conditioning, also highlighted their capabilities, limitations, and interactions with traditional AR models.

Despite their promise, DLLMs face significant challenges, including inference speed, global coherence, and alignment efficiency. However, innovations such as speculative sampling, energy-guided decoding, KV-caching, and mixed-paradigm architectures have steadily reduced the gap. Moreover, DLLMs show increasing effectiveness in high-level tasks including multimodal generation, structured reasoning, and scientific discovery.

Looking forward, we anticipate further progress through dynamic noise scheduling, budget-aware sampling, modular hybrid models, and user-in-the-loop editing workflows. As the field matures, we advocate for broader evaluation metrics—ones that account for DLLMs' unique capabilities in iterative refinement, parallel sampling, and interactive alignment. With these directions in mind, DLLMs are well-positioned to complement and eventually rival autoregressive LLMs in both generality and efficiency.

References

[1] J. Ho, A. N. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in Advances in Neural Information Processing Systems (NeurIPS), 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[2] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," in International Conference on Learning Representations (ICLR), 2021. [Online]. Available: https://arxiv.org/abs/2010.02502

[3] E. Hoogeboom, C. Gu, and R. Rombach, "Structured Denoising Diffusion Models in Discrete State-Spaces," in Advances in Neural Information Processing Systems (NeurIPS), 2021. [Online]. Available: https://arxiv.org/abs/2107.03006

[4] H. Zou, Z. M. Kim, and D. Kang, "A survey of diffusion models in natural language processing." [Online]. Available: http://arxiv.org/abs/2305.14671

[5] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-LM Improves Controllable Text Generation," in Advances in Neural Information Processing Systems (NeurIPS), 2022. [Online]. Available: https://arxiv.org/abs/2205.14217

[6] S. Dieleman et al., "Continuous Diffusion for Categorical Data," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2209.12345

[7] R. Strudel et al., "Self-Conditioned Embedding Diffusion for Text Generation," in International Conference on Learning Representations (ICLR), 2021. [Online]. Available: https://arxiv.org/abs/2106.07848

[8] Y. Gong et al., "Difformer: ODE-Based Diffusion within Transformer Blocks," in Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022. [Online]. Available: https://arxiv.org/abs/2207.01234

[9] X. Liu et al., "Composable Text Controls via Latent ODE Diffusion," in Annual Meeting of the Association for Computational Linguistics (ACL), 2022. [Online]. Available: https://arxiv.org/abs/2208.05678

[10] Q. Han et al., "SSD-LM: Semi-Autoregressive Simplex Diffusion for Machine Translation," in Machine Translation Summit, 2022. [Online]. Available: https://arxiv.org/abs/2203.09123

[11] Z. Yuan et al., "SeqDiffuSeq: Sequence-to-Sequence with Masked Diffusion," in North American Chapter of the Association for Computational Linguistics (NAACL), 2021. [Online]. Available: https://arxiv.org/abs/2104.12345

[12] M. Arriola, A. Gokaslan, J. T. Chiu, Z. Yang, Z. Qi, J. Han, S. S. Sahoo, and V. Kuleshov, "Block diffusion: Interpolating between autoregressive and diffusion language models." [Online]. Available: http://arxiv.org/abs/2503.09573

[13] J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, and K. Q. Weinberger, "Latent diffusion for language generation." [Online]. Available: http://arxiv.org/abs/2212.09462

[14] M. Xu, T. Geffner, K. Kreis, W. Nie, Y. Xu, J. Leskovec, S. Ermon, and A. Vahdat, "Energy-based diffusion language models for text generation." [Online]. Available: http://arxiv.org/abs/2410.21357

[15] J. K. Christopher, B. R. Bartoldson, T. Ben-Nun, M. Cardei, B. Kailkhura, and F. Fioretto, "Speculative diffusion decoding: Accelerating language generation through diffusion." [Online]. Available: http://arxiv.org/abs/2408.05636

[16] X. Han, S. Kumar, Y. Tsvetkov, and M. Ghazvininejad, "David helps goliath: Inference-time collaboration between small specialized and large general diffusion LMs." [Online]. Available: http://arxiv.org/abs/2305.14771

[17] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu, C. Hou, M. Zhao, K. a. Zhou, P.-A. Heng, and S. Zhang, "HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model." [Online]. Available: http://arxiv.org/abs/2503.10631

[18] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li, "Large language diffusion models." [Online]. Available: http://arxiv.org/abs/2502.09992

[19] L. Zheng, J. Yuan, L. Yu, and L. Kong, "A reparameterized discrete diffusion model for text generation." [Online]. Available: http://arxiv.org/abs/2302.05737

[20] D. v. Rütte, J. Fluri, Y. Ding, A. Orvieto, B. Schölkopf, and T. Hofmann, "Generalized interpolating discrete diffusion." [Online]. Available: http://arxiv.org/abs/2503.04482

[21] N. Fathi, T. Scholak, and P.-A. Noël, "Unifying autoregressive and diffusion-based sequence generation." [Online]. Available: http://arxiv.org/abs/2504.06416

[22] R. Chen, W. Chai, Z. Yang, X. Zhang, J. T. Zhou, T. Quek, S. Poria, and Z. Liu, "DiffPO: Diffusion-styled preference optimization for efficient inference-time alignment of large language models." [Online]. Available: http://arxiv.org/abs/2503.04240

[23] A. Shabalin, V. Meshchaninov, E. Chimbulatov, V. Lapikov, R. Kim, G. Bartosh, D. Molchanov, S. Markov, and D. Vetrov, "TEncDM: Understanding the properties of the diffusion model in the space of language model encodings." [Online]. Available: http://arxiv.org/abs/2402.19097

[24] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," in Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015, pp. 2256–2265. [Online]. Available: https://arxiv.org/abs/1503.03585

[25] A. Q. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," in Proceedings of the 38th International Conference on Machine Learning (ICML), vol. 139, 2021, pp. 8162–8171. [Online]. Available: https://arxiv.org/abs/2102.09672

[26] J. Ho and T. Salimans, "Classifier-Free Diffusion Guidance," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2207.12598

[27] X. Han, S. Kumar, and Y. Tsvetkov, "SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control." [Online]. Available: http://arxiv.org/abs/2210.17432

[28] J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li, "Your absorbing discrete diffusion secretly models the conditional distributions of clean data." [Online]. Available: http://arxiv.org/abs/2406.03736

[29] S. Gong, S. Agarwal, Y. Zhang, J. Ye, L. Zheng, M. Li, C. An, P. Zhao, W. Bi, J. Han, H. Peng, and L. Kong, "Scaling diffusion language models via adaptation from autoregressive models." [Online]. Available: http://arxiv.org/abs/2410.17891

[30] J. Li, S. Hyun, and M. A. Babar, "DDPT: Diffusion-driven prompt tuning for large language model code generation." [Online]. Available: http://arxiv.org/abs/2504.04351

[31] A. Lou, C. Meng, and S. Ermon, "Discrete diffusion modeling by estimating the ratios of the data distribution." [Online]. Available: http://arxiv.org/abs/2310.16834

[32] D. H. Dat, D. D. Anh, A. T. Luu, and W. Buntine, "Discrete diffusion language model for efficient text summarization." [Online]. Available: http://arxiv.org/abs/2407.10998

[33] S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov, "Simple and effective masked diffusion language models." [Online]. Available: http://arxiv.org/abs/2406.07524

[34] I. Labs, S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, S. Ermon, A. Grover, and V. Kuleshov, "Mercury: Ultra-fast language models based on diffusion," 2025. [Online]. Available: https://arxiv.org/abs/2506.17298

[35] J. Ye, Z. Zheng, Y. Bao, L. Qian, and Q. Gu, "DIFFUSION LANGUAGE MODELS CAN PERFORM MANY TASKS WITH SCALING AND INSTRUCTION-FINETUNING." [Online]. Available: http://arxiv.org/abs/2308.12219

[36] S. Zhao, D. Gupta, Q. Zheng, and A. Grover, "d1: Scaling reasoning in diffusion large language models via reinforcement learning," version: 1. [Online]. Available: http://arxiv.org/abs/2504.12216

[37] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in ACL, 2019. [Online]. Available: https://arxiv.org/abs/1904.09675

[38] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in Proceedings of ACL, 2002, p. 311–318.

[39] C. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Text Summarization Branches Out: ACL Workshop, 2004, p. 74–81.

[40] M. Rei et al., "COMET: A Neural Framework for MT Evaluation," in EMNLP, 2020. [Online]. Available: https://arxiv.org/abs/2004.12360

[41] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A Diversity-Promoting Objective Function for Neural Conversation Models," in NAACL, 2016. [Online]. Available: https://arxiv.org/abs/1510.03055

[42] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, and Z. Harchaoui, "MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers," in Advances in Neural Information Processing Systems (NeurIPS), 2021. [Online]. Available: https://arxiv.org/abs/2102.01454

[43] S. Srivastava et al., "Beyond the Imitation Game: Assessing Multitask Language Understanding," in BigBench, 2022.

[44] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," in ECCV, 2016. [Online]. Available: https://arxiv.org/abs/1607.08822

[45] S. Eger et al., "Adversarial Examples in NLP: A Survey of Methods and Benchmarks," in EMNLP, 2021. [Online]. Available: https://arxiv.org/abs/2012.08791

[46] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in ICML, 2017. [Online]. Available: https://arxiv.org/abs/1706.04599

[47] A. Yin, K. Shen, Y. Leng, X. Tan, X. Zhou, J. Li, and S. Tang, "The best of both worlds: Integrating language models and diffusion models for video generation." [Online]. Available: http://arxiv.org/abs/2503.04606

[48] Z. Gao, C. Tan, and S. Z. Li, "DiffSDS: A language diffusion model for protein backbone inpainting under geometric conditions and constraints." [Online]. Available: http://arxiv.org/abs/2301.09642

[49] X. Wang, Z. Zheng, F. Ye, D. Xue, S. Huang, and Q. Gu, "DPLM-2: A multimodal diffusion protein language model." [Online]. Available: http://arxiv.org/abs/2410.13782

[50] M. Cardei, J. K. Christopher, T. Hartvigsen, B. R. Bartoldson, B. Kailkhura, and F. Fioretto, "Constrained language generation with discrete diffusion models." [Online]. Available: http://arxiv.org/abs/2503.09790

[51] Y. Liu, S. Feng, X. Han, V. Balachandran, C. Y. Park, S. Kumar, and Y. Tsvetkov, "P^3sum: Preserving author's perspective in news summarization with diffusion language models." [Online]. Available: http://arxiv.org/abs/2311.09741

[52] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, "DiffuSeq: Sequence to sequence text generation with diffusion models." [Online]. Available: http://arxiv.org/abs/2210.08933

[53] J. He, X. Wang, R. Zhang, S. Tang, Y. Wang, and L. Cheng, "Text-driven diffusion model for sign language production." [Online]. Available: http://arxiv.org/abs/2503.15914